

УДК 004.584

ИСПОЛЬЗОВАНИЕ ЧАТ-БОТА ДЛЯ АВТОМАТИЗАЦИИ ПРОЦЕССА СБОРА ОБРАТНОЙ СВЯЗИ ДЛЯ СОТРУДНИКОВ ПРЕДПРИЯТИЯ

Бер В.А., студент гр. ИТб-201, IV курс
Протоdjяконов А.В., доцент (к.н.) кафедры ИиАПС
Кузбасский государственный технический университет имени
Т. Ф. Горбачева
г. Кемерово

В современном динамичном мире бизнеса, где конкуренция постоянно растет, и технологии развиваются с высокой скоростью, важность обратной связи от сотрудников становится все более очевидной. Сотрудники, находящиеся на передовой бизнес-процессов, часто первыми сталкиваются с проблемами и неэффективностью в работе, и их взгляды могут предоставить ценные инсайты для улучшения операций и стратегий компании.

Обратная связь помогает не только выявлять слабые места, но и способствует вовлеченности персонала, создавая атмосферу открытости и доверия. Сотрудники, которые видят, что их мнение ценится и приводит к реальным изменениям, более мотивированы и лояльны к компании.

Однако сбор обратной связи может быть сложной задачей, особенно в крупных организациях с множеством отделов и сложными иерархиями. Традиционные методы, такие как опросы и собрания, могут быть время затратными и не всегда эффективными. В этом контексте телеграмм-боты представляют собой инновационное решение, позволяющее собирать обратную связь быстро, конфиденциально и в удобной форме.

Телеграмм, как платформа, обладает рядом преимуществ, включая высокую скорость доставки сообщений, шифрование данных и широкое распространение среди пользователей. Создание специализированного бота для сбора обратной связи позволяет сотрудникам делиться своими мыслями и предложениями в любое время и из любого места, что делает процесс максимально ненавязчивым и удобным.

В данном докладе мы рассмотрим, как можно создать телеграмм-бота, который будет эффективно собирать, обрабатывать и анализировать обратную связь от сотрудников, обеспечивая руководству компании ценные данные для принятия обоснованных управленческих решений.

Чат-бот будет работать в мессенджере Telegram, и написан на языке Python с библиотекой python-telegram-bot. Также в систему будет входить база данных (БД), к которой будет обращаться бот, созданная в СУБД Microsoft SQL Server, и хранящая контактные данные сотрудника и его оценку качества работы и замечания.

Программа, управляющая чат-ботом, будет размещена на сервере и работать в режиме 24/7, обрабатывая запросы от сотрудников. Бот будет предоставлять интерфейс для анонимного ввода оценок и комментариев относительно работы различных отделов компании.

Для взаимодействия с ботом сотрудникам будет предложено отправить команду с номером телефона, который они указали при регистрации в базе данных компании. Это позволит чат-боту связать пользователя с его учетной записью, либо выбрать анонимную отправку оценки. Далее пользователь выбирает отдел, ставит оценку и пишет комментарий (необязательно). После этого бот показывает введенные данные и просит подтвердить отправку отзыва. Если пользователь подтверждает отправку, бот сохраняет отзыв в БД с учетом выбранного типа анонимности, если пользователь не подтверждает отправку, то происходит завершение процесса. Диаграмма деятельности процесса оповещения представлена на рисунке 1.

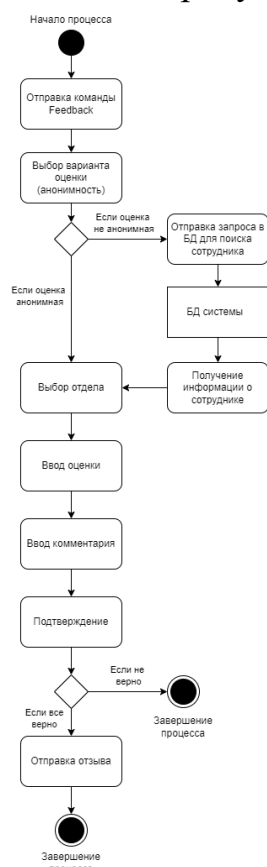


Рисунок 1 – Диаграмма деятельности процесса оповещения.

Чат-боты в Telegram не могут первыми начать переписку с пользователем, чтобы предотвратить распространение спама. Поэтому необходимо соотнести сотрудника, запись о котором уже есть в БД, с пользователем, который добавил чат-бота. Для этого в приветственном сообщении попросим пользователя отправить команду с номером телефона, который он указал при добавлении записи о нем в БД. После этого, чат-бот обратится к БД с запросом на поиск сотрудника с таким номером телефона и последующей записью идентификатора чата в атрибут кортежа данного сотрудника. Если данный запрос завершился успехом, то в ответ пользователю должно прийти подтверждающее сообщение, обращающееся к пользователю по имени и отчеству. В противном случае, пользователю должно прийти сообщение с текстом, что сотрудник с таким номером телефона не найден.

Пользователь начинает взаимодействие с ботом, отправляя команду /start. Бот предлагает пользователю выбрать, хочет ли он продолжить анонимно или идентифицировать себя. Пользователь делает выбор. Если выбрана анонимность, бот информирует пользователя о том, что его данные не будут сохранены и он может свободно использовать бота для отправки анонимных отзывов и следует переход к завершению процесса идентификации. Если выбрана идентификация, Бот просит пользователя отправить свой номер телефона в формате команды (например, /auth 1234567890). Бот обращается к базе данных, чтобы найти сотрудника с указанным номером телефона. После проверки БД возвращает результат поиска. При успешной идентификации бот сохраняет идентификатор чата пользователя для дальнейшего взаимодействия и отправляет приветственное сообщение, обращаясь к сотруднику по имени и отчеству. Если сотрудник не найден, бот отправляет сообщение о том, что номер телефона не найден в базе данных и предлагает повторить попытку. Процесс идентификации завершается, и пользователь может продолжить использовать бота в соответствии с выбранным режимом (анонимным или идентифицированным).

Диаграмма деятельности для идентификации пользователя представлена на рисунке 2.

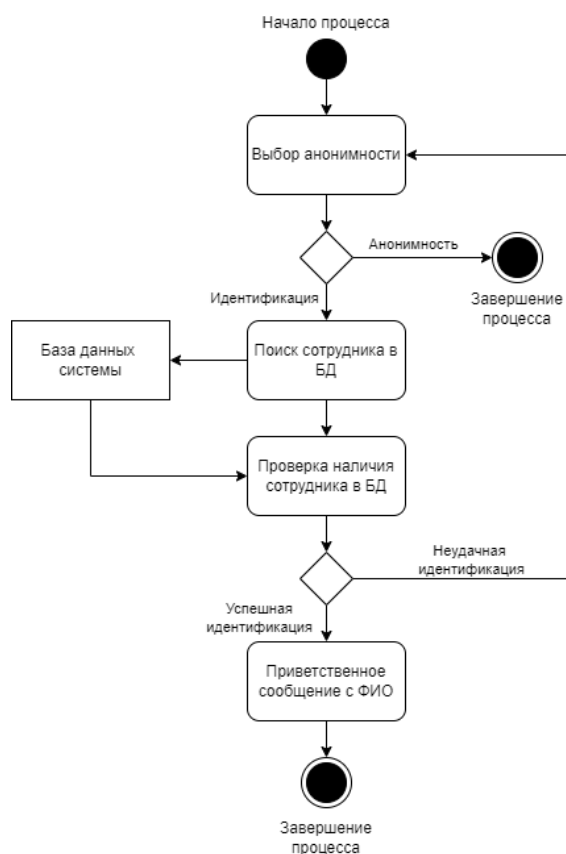


Рисунок 2 – Диаграмма деятельности процесса идентификации пользователя.

Для того чтобы начать работу с чат-ботом в Telegram, первым делом потребуется его зарегистрировать. Этот процесс осуществляется через официального представителя Telegram для ботов — BotFather. С его помощью вы сможете задать имя вашего бота, которое будет видно пользователям, а также уникальный идентификатор для поиска бота в системе, который должен заканчиваться на "bot" (например, @Chat_bot). Кроме того, BotFather позволяет настроить ряд внешних и функциональных параметров вашего бота.

Взаимодействие с BotFather происходит посредством команд, аналогично общению с любыми другими ботами в Telegram. Вот некоторые из команд, которые доступны в BotFather:

- /newbot — команда для создания нового бота;
- /setname — команда для изменения имени бота;
- /setcommands — команда для настройки списка команд, доступных боту;
- /setuserpic — команда для изменения аватара бота;
- /setabouttext — команда для добавления описания бота;
- /token — команда для получения токена, необходимого для управления ботом через разработанное приложение.

После того, как вы зададите имя и уникальный идентификатор бота, BotFather предоставит вам токен — уникальную строку символов, которая будет использоваться для подключения и управления ботом через ваше приложение. Важно обеспечить конфиденциальность этого токена, так как он является ключом к управлению вашим ботом. Созданный бот представлен на рисунке 3.

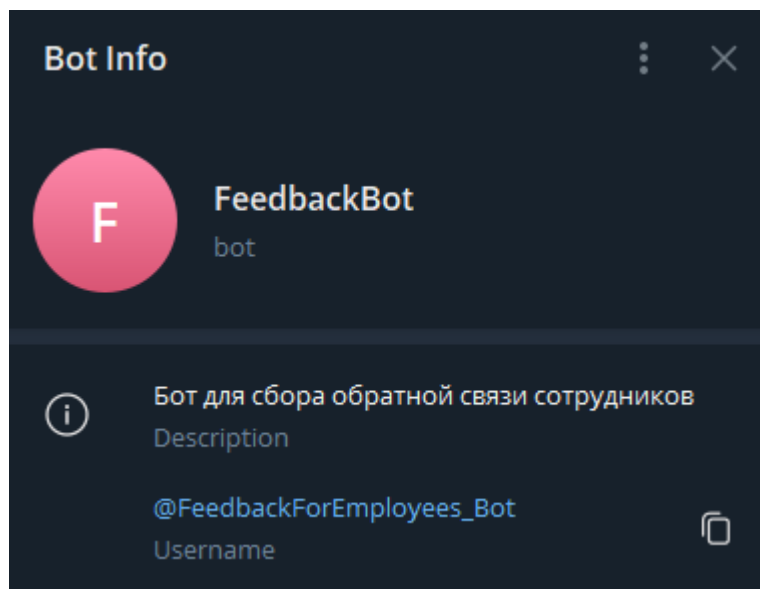


Рисунок 3 – Созданный чат-бот в telegram.

Для начала работы с чат-ботом в Telegram необходимо установить библиотеку `python-telegram-bot`, которая может быть установлена через систему управления пакетами `pip`. Это позволит нам использовать все необходимые классы и методы для взаимодействия с Telegram API.

Приложение, работающее на сервере, может быть реализовано как консольное приложение с возможностью ввода токена, полученного от BotFather. В качестве альтернативы можно создать простой интерфейс с полем ввода и кнопкой для запуска бота.

Telegram предоставляет два способа получения обновлений от бота: через метод `get_updates()` или с помощью вебхуков. В нашем случае мы будем использовать метод `get_updates()`, который будет находиться в бесконечном цикле для постоянного получения обновлений. Этот метод проще в реализации, хотя и может создавать дополнительную нагрузку на сервера Telegram.

Чтобы обработка запросов от пользователей не блокировала основной поток приложения, мы используем механизм многопоточности,

предоставляемый Python. Это позволит нам обрабатывать запросы в фоновом режиме.

Для подключения к чат-боту создадим экземпляр класса Updater и передадим в него токен. Подключение чат-бота показано на рисунке 4.

```
from telegram.ext import Updater

updater = Updater(token='YOUR_BOT_TOKEN', use_context=True)
dispatcher = updater.dispatcher
```

Рисунок 4 – Подключение к чат-боту.

Чат-бот будет отвечать на следующие команды:

- /feedback – Стартовая команда, после которой будет предложен выбор анонимности.
- /anonymity – Выбор анонимности.
- /rating – Ввод оценки.
- /skip_comment – Пропуск ввода комментария.
- /comment – Ввод комментария.
- /confirmation – Подтверждение.
- /submit_feedback – Отправка отзыва.
- /cancel – Завершение разговора.

Пример обработки команды /feedback показан на рисунке 5.

```
def feedback(update: Update, context: CallbackContext):
    reply_keyboard = [['Анонимно', 'Не анонимно']]
    update.message.reply_text(
        'Хотите оставить анонимный отзыв или нет?',
        reply_markup=ReplyKeyboardMarkup(reply_keyboard, one_time_keyboard=True)
    )
```

Рисунок 5 – Обработка команды /start.

В Telegram можно сделать два вида кнопок - Inline кнопки, отображающиеся под отправленным сообщением чат-бота, и Reply кнопки, отображающиеся вместо клавиатуры у пользователя. Вывод Inline кнопки показан на рисунке 6.

```
def confirmation(update: Update, context: CallbackContext):  
    reply_markup = InlineKeyboardMarkup([  
        [InlineKeyboardButton("Подтвердить", callback_data='confirm')],  
        [InlineKeyboardButton("Отменить", callback_data='cancel')]  
    ])
```

Рисунок 6 – Вывод Inline кнопки.

Тестирование чат-бота показано на рисунке 7

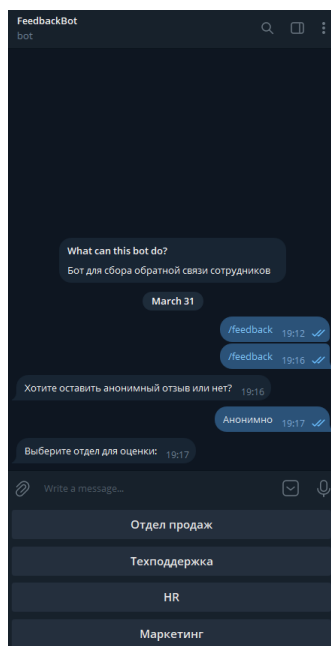


Рисунок 7 – Тестирование чат-бота.

Внедрение телеграмм-бота для анонимного сбора обратной связи позволит сотрудникам высказываться свободно и без страха перед последствиями. Это способствует созданию доверительной атмосферы и позволяет руководству компании получать более честные и конструктивные отзывы. Использование Python и библиотеки python-telegram-bot обеспечит гибкость и масштабируемость решения, а также упростит интеграцию с существующими системами и базами данных. Благодаря этому нововведению, компания сможет оперативно реагировать на предложения сотрудников и улучшать внутренние процессы, что в конечном итоге приведет к повышению общей продуктивности и удовлетворенности персонала.

Список литературы:

1. Документация Telegram: Боты. [Электронный ресурс] URL: <https://tlgrm.ru/docs/bots> (дата обращения: 26.03.2024).

2. Библиотека python-telegram-bot в Python. [Электронный ресурс] URL: <https://tlgrm.ru/docs/bots/api> (дата обращения: 23.03.2024).
3. Моделирование на языке UML в среде Visual Paradigm 14. [Электронный ресурс] – URL: <http://sp.cs.msu.ru/ooap/exer2017.html#exer34> (дата обращения: 23.03.2024).
4. Шумилина М.А., Коробко А.Н. РАЗРАБОТКА ЧАТ-БОТА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON В МЕССЕНДЖЕРЕ "TELEGRAM". [Электронный ресурс] // Научные известия. 2022. №28. URL: <https://cyberleninka.ru/article/n/razrabotka-chat-bota-na-yazyke-programmirovaniya-python-v-messendzhere-telegram> (дата обращения: 23.03.2024).